

Scientific journal
PHYSICAL AND MATHEMATICAL EDUCATION
Has been issued since 2013.

ISSN 2413-158X (online)
ISSN 2413-1571 (print)

Науковий журнал
ФІЗИКО-МАТЕМАТИЧНА ОСВІТА
Видається з 2013.

<https://fmo-journal.org/>



Кобильник Т.П., Когут У.П., Жидик В.Б. Методичні аспекти вивчення основ алгоритмізації і програмування мовою Python у шкільному курсі інформатики у старших класах. Фізико-математична освіта, 2021. Випуск 5(31). С. 36-44.

Kobylnyk T., Kohut U., Zhydyk V. Methodical aspects of studying the fundamentals of algorithmization and programming language Python school course in informatics in high school. Physical and Mathematical Education, 2021. Issue 5(31). P. 36-44.

DOI 10.31110/2413-1571-2021-031-5-006
УДК 378:147:51:004

Т.П. Кобильник

Національний університет «Львівська політехніка», Україна
taras.p.kobylnyk@lpnu.ua
<https://orcid.org/0000-0002-2703-7570>

У.П. Когут

Дрогобицький державний педагогічний університет імені Івана Франка, Україна
ulyana3001@gmail.com
<https://orcid.org/0000-0002-2861-2274>

В.Б. Жидик

Дрогобицький державний педагогічний університет імені Івана Франка, Україна
Zhvb63@gmail.com
<https://orcid.org/0000-0002-9876-0053>

МЕТОДИЧНІ АСПЕКТИ ВИВЧЕННЯ ОСНОВ АЛГОРИТМІЗАЦІЇ І ПРОГРАМУВАННЯ МОВОЮ PYTHON У ШКІЛЬНОМУ КУРСІ ІНФОРМАТИКИ У СТАРШИХ КЛАСАХ

АНОТАЦІЯ

Основи алгоритмізації і програмування є однією з основних змістових ліній курсу інформатики у школі. У статті досліджено проблеми вивчення елементів основ алгоритмізації та програмування в шкільному курсі інформатики у 10-11 класах.

Формулювання проблеми. Важливою педагогічною проблемою є формування і розвиток алгоритмічних навичок учнів. Проблеми вдосконалення методичної системи навчання інформатики в загальній середній школі потребують подальшого дослідження з метою пошуку більш ефективних форм, методів і засобів навчання. Необхідно забезпечити формування зв'язку між теоретичною і прикладною інформатикою, знати способи застосування інформаційних технологій для розв'язування різноманітних задач. А також визначено переваги мови Python у навчанні інформатики та особливості їх педагогічного застосування у навчанні основ алгоритмізації та програмування.

Матеріали і методи. Для досягнення поставленої мети застосовано комплекс методів: системного аналізу наукових праць, що охоплюють проблему дослідження, аналіз вітчизняного досвіду вивчення елементів програмування в шкільному курсі інформатики; узагальнення та обґрунтування теоретичних засад використання мови програмування Python в шкільному курсі інформатики.

Результати. Розглянуто основні характеристики мови програмування Python та наведено можливості її використання у процесі навчання основ алгоритмізації в шкільному курсі інформатики. Вивчення основ алгоритмізації та програмування є одним із засобів формування операційного стилю мислення. Знання даної змістової лінії передбачає здатність формалізувати задачу, визначити у ній зв'язки між частинам, обирати якнайкращий алгоритм розв'язування завдання, вміння правильно інтерпретувати та аналізувати отримані результати.

Висновки. Мова Python – одна з найпопулярніших мов програмування. Вона знаходить своє застосування у різних галузях. Відповідно ринок праці потребує фахівців, які могли створювати програмні продукти мовою Python. Заклади загальної середньої освіти не залишилися осторонь цього процесу. Автори шкільних підручників починають включати мову Python до їх змісту. Таким чином, виникає потреба у розробці навчально-методичних матеріалів з вивчення елементів алгоритмізації і програмування мовою Python. Методологічне значення цього розділу шкільного курсу інформатики полягає у розкритті важливості алгоритмів.

КЛЮЧОВІ СЛОВА: заклад загальної середньої освіти, шкільний курс інформатики, основи алгоритмізації, програмування, мова програмування Python.

ВСТУП

Постановка проблеми. Інформатика у старших класах вивчається на рівні стандарту та на профільному рівні.

Програма (рівень стандарту) (Інформатика 10-11 клас, 2020) розрахована на вивчення інформатики у 10-11 класах як вибірково-обов'язковий предмет, 35 годин якого складає базовий модуль. Аналіз змісту базового модуля показав, що не передбачено вивчення жодних розділів, пов'язаних з алгоритмами, алгоритмізацією та програмуванням. Проте програмою передбачено вибірково-обов'язковий модуль «Креативне програмування», який пов'язаний з програмуванням.

Слід зазначити, що вибірково-обов'язковий (один чи кілька) модуль учителю рекомендується обирати відповідно до профілю закладу загальної середньої освіти, індивідуальних інтересів і здібностей учнів, матеріально-технічної бази та наявного програмного забезпечення. Оскільки модуль «Креативне програмування» є вибірково-обов'язковим, то не всі учні, які навчаються на рівні стандарту, будуть мати змогу вивчати елементи алгоритмізації та програмування.

Програмою з інформатики (профільне навчання) для учнів 10-11 класів (Інформатика для 10-11 класів (профільне навчання), 2020) передбачено вивчення таких розділів як «Мова програмування та структури даних», «Алгоритми» та «Парадигми програмування», які так чи інакше передбачають опанування певною мовою програмування. Слід зазначити, що вивчення основ програмування, мови, синтаксису, структури та правил написання програм передбачено у розділі «Мова програмування та структури даних».

Навчальними програмами з інформатики не визначено певної мови програмування для вивчення основ алгоритмізації. Тому вибір залежить від вчителя або закладу освіти. Проте автори шкільних підручників починають включати мову Python до їх змісту. Проте методичних рекомендацій щодо використання мови Python у шкільному курсі інформатики є мало, на відміну від Pascal. Таким чином, виникає потреба у розробці навчально-методичних матеріалів з вивчення елементів алгоритмізації і програмування мовою Python.

Аналіз актуальних досліджень. У підручниках з інформатики для 5-6 класів елементи алгоритмізації та програмування пропонується вивчати з використанням мови Scratch. Зауважимо, що у шкільних підручниках з інформатики (2016 року) для учнів 7-9 класів мова Python не була відображена, за винятком підручників авторського колективу Н.В. Морзе. У всіх підручниках як основну мову програмування було обрано Free Pascal і середовище Lazarus. У підручниках (Морзе& Барна& Вембер, 2016) разом зі згаданими середовищем і мовою програмування подаються відомості і про Python з прикладами створення програм. Зокрема, вивчення у 8-му та 9-му класах розділів «Алгоритми роботи з об'єктами та величинами» та «Табличні величини та алгоритми їх опрацювання» відповідно проводиться з використанням мови Python.

Ситуація у підручниках 2020 року випуску для 7-9 класів кардинально змінилася. Автори відмовляються від мови Free Pascal середовища Lazarus. Натомість зосередилися на описі мови Python. Наприклад, у підручниках (Морзе&Барна, 2020) для 7-го класу при вивченні розділу «Алгоритми і програми» поряд з Python подаються відомості і про Scratch. Також зауважимо, що у підручнику для 8-го класу (Ривкінд& Лисенко&Чернікова&Шакотько В.В., 2021) надалі для вивчення розділу «Алгоритми і програми» пропонується і Lazarus, і Python.

Якщо брати до уваги старшу школу, то, як вже було сказано, на рівні стандарту вивчення програмування можливе тільки у вибірково-обов'язковому модулі «Креативне програмування». Відповідно підручників, де були відображені навчальні матеріали з даного модуля нема.

У підручнику (Руденко&Речич&Потієнко, 2019) для 10 класу профільного рівня у розділі «Мови програмування та структури даних» як мову програмування обрано Python. Зокрема, у ньому поряд з основами алгоритмізації та програмування подаються відомості й про основи об'єктно-орієнтованого програмування. Це здійснюється на прикладі мови Python і середовища програмування IDLE. Зокрема, у підручнику розглядаються такі питання як: структура і способи виконання проектів мовою Python; оператори, вирази, і засоби опрацювання чисел; реалізація базових алгоритмічних конструкцій; вбудовані типи даних та їх опрацювання; функції користувача та модулі Python; основи об'єктно-орієнтованого програмування; основи графічного інтерфейсу користувача.

У підручнику (Руденко&Речич&Потієнко, 2019) автори для 11 класу профільного рівня у розділі «Алгоритми» пропонують вивчати і реалізовувати базові алгоритми (сортування та пошуку даних, опрацювання рядків, графі, динамічне програмування і жадібні алгоритми, основи обчислювальної геометрії), з використанням мови Python та середовища IDLE.

Ще у 2013 році редакція журналу «Комп'ютер у школі та сім'ї» звернулася до відомих учителів інформатики, фахівців у галузі навчальної інформатики з проханням взяти участь в обговоренні питання «Яку мову програмування вивчати у школі?». Таке обговорення вилилося у кілька публікацій, у яких знані педагоги (та й не тільки педагоги) висловлювали свої думки з приводу цього питання. У кожного з них є своя обґрунтована позиція щодо вивчення мови програмування у школі: Pascal, C, C++, Visual Basic, Visual Basic For Application, JavaScript, Python тощо. Підсумовуючи, можна зробити висновок, що усі аргументи на користь тієї чи іншої мови програмування є важливими, але останнє слово завжди залишається за вчителем.

На думку авторів статті (Юрченко&Семеніхіна&Хворостіна&Удовиченко&Петренко, 2019), перспективним сьогодні в старшій школі вивчати популярні мови Java, Python та C++.

Мета статті. З огляду на це метою статті є висвітлення підходів вивчення основ алгоритмізації та програмування мовою Python в шкільному курсі інформатики у старших класах.

МЕТОДИ ДОСЛІДЖЕННЯ

Для досягнення поставленої мети застосовано комплекс методів: системного аналізу наукових праць, що охоплюють проблему дослідження, вивчення вітчизняного досвіду вивчення елементів програмування в шкільному курсі інформатики; узагальнення та обґрунтування теоретичних засад використання мови програмування PYTHON в шкільному курсі інформатики.

РЕЗУЛЬТАТИ ДОСЛІДЖЕННЯ

Щоб обрати певну мову програмування, треба визначити критерії, яким повинна вона відповідати.

Обрана мова для вивчення програмування у школі повинна відповідати таким вимогам (Комп'ютер у школі та сім'ї, 2013): транслятор мови повинен бути кросплатформним і безкоштовним; мова повинна мати простий синтаксис, з одного боку, і потужні засоби, з іншого; програми повинні бути короткими і зрозумілими, форма запису повинна бути максимально звичною для людини; мати розвинуті бібліотеки, можливості розробки різноманітних додатків; повинна підтримувати стилі як структурного, так і функціонального та об'єктно-орієнтованого програмування; мова повинна бути сучасною, не «мертвою», входити хоча б у 10 найбільш уживаних в реальному виробництві мов програмування; середовище розробки програм не повинно бути переважане додатковими функціями і можливостями, складність освоєння яких «затмарює» і відсуває на другий план алгоритмічну складність проекту. Обов'язковою також є наявність консольного транслятора.

Таким вимогам відповідає мова Python (<http://www.python.org>), яку використовують не тільки в навчальних цілях. Python займає друге місце (після C) в рейтингу популярності мов програмування TIOBE Programming Community Index (TIOBE Index, 2020) (станом на вересень 2021 року). мова активно розвивається, має відкритий (Open Source) програмний код, поширюється за GPL-ліцензією.

У (Комп'ютер у школі та сім'ї №8, с. 18) Пасіхов Ю.Я. та Кравець Г.П. наводять деякі переваги мови Python над іншими (C, C++, Pascal), серед яких слід відзначити такі. Прості програми записуються у кілька рядків, відсутні інструкції, що не мають безпосереднього відношення до алгоритму (наприклад, `int main()` у мові C чи C++). Як правило, програми мовою Python є коротшими, ніж на C, C++, Pascal, та C#. мова сучасна, підтримує високорівневі складені структури даних (списки, множини, асоціативні масиви, інше). мова Python із самого початку створювалася на основі парадигми об'єктно-орієнтованого програмування, але чудово є пристосованою для структурного і функціонального програмування.

Крім консольного інтерпретатора мови, Python має кілька різних середовищ розробки програм, зокрема IDLE (стандартне середовище, що входить у дистрибутив, є простим і зрозумілим). Слід зауважити, що детальний аналіз середовищ програмування мовою Python наведено у (Базурін, 2018). Там же автор зазначає, що до найбільш популярних середовищ програмування мовою Python належать: IDLE, PyCharm, PyDEV, WingWare, Komodo IDE, Eric, Eclipse, Geany, Spyder, PyScripter. Крім цього, також використовуються online середовища.

Освоєння будь-якої мови програмування розпочинається з вивчення її синтаксису та алфавіту, з якого створюються лексеми (ключові слова, літерали, ідентифікатори, операції, знаки пунктуації). Зауважимо, що мова Python є доступною для використання у всіх поширених сучасних операційних системах сімейств Windows, Linux, Mac. Як і будь-яка мова, Python має свої особливості, серед яких виокремимо так:

- ✓ наявність інтерактивного режиму та підтримка файлу програмного коду;
- ✓ розрізняється регістр введених літер, тобто A та a – це різні змінні (чи символи);
- ✓ відступи є обов'язковими (їх використовують для задання певного блоку операторів);
- ✓ динамічна типізація змінних, тобто у програмі не потрібно описувати змінні (до якого типу вони належать);
- ✓ змінні зберігають адресу об'єкта у пам'яті комп'ютера, а не сам об'єкт;
- ✓ відсутнє поняття масиву: використовуються списки;
- ✓ перед першим запуском програмного коду його слід зберегти. Якщо цього не виконати, то користувачу запропонують це зробити;

✓ наявність значної кількості бібліотек. Наприклад, бібліотека `math`, яка містить основні математичні функції, для використання яких необхідно виконати `import math`. Далі використовувати функції цієї бібліотеки так, наприклад `sin(x)` чи `sqrt(x)`: `math.sin(x)` чи `math.sqrt(x)` відповідно.

Слід зазначити, що учні 10-х класів уже знайомі основами алгоритмізації та програмування, які вони вивчали у 5-9 класах з використанням мов Scratch, Free Pascal, рідше Python. Тому ми не будемо зупинятися на ознайомленні з алгоритмічною структурою слідування у мові Python. Перейдемо відразу до алгоритмічної структури розгалуження з метою узагальнення і систематизації знань з даної теми, отриманих раніше.

Пояснення умовного оператора `if` пропонується проводити на таких задачах як «Визначення чверті, до якої належить точка», «Розв'язування лінійного рівняння» та «Розв'язування квадратного рівняння». Це зумовлено тим, що для знаходження мінімального та максимального з двох чи трьох (або більше) чисел у мові Python можна використовувати функції `min` та `max` відповідно.

Наведемо приклад розв'язування лінійного рівняння.

Приклад 1. Скласти програму, за якою отримується розв'язок лінійного рівняння $ax+b=0$, де a, b – деякі параметри, x – невідома.

Перш за все учнів треба привчати до складання алгоритму розв'язування задачі, а потім вже до його реалізації мовою програмування.

Спочатку розв'яжемо дане рівняння аналітично. Або іншими словами опишемо алгоритм словесно-аналітично. Розв'язати рівняння з параметрами означає, що для всіх можливих значень a, b необхідно вказати можливі значення невідомої x .

1. Ввести значення a, b .
2. Якщо $a \neq 0$, то виконати п.3, інакше п.5.
3. Обчислити $x = -b/a$. Вивести значення x .
4. Перейти до п.9.
5. Якщо $b \neq 0$, то виконати п.6, інакше п.8.
6. Вивести повідомлення «Рівняння коренів не має».
7. Перейти до п.9.
8. Вивести повідомлення «Рівняння має безліч коренів».

9. Кінець.

На основі словесно-аналітичного опису алгоритму складемо блок-схему (Рис. 1):

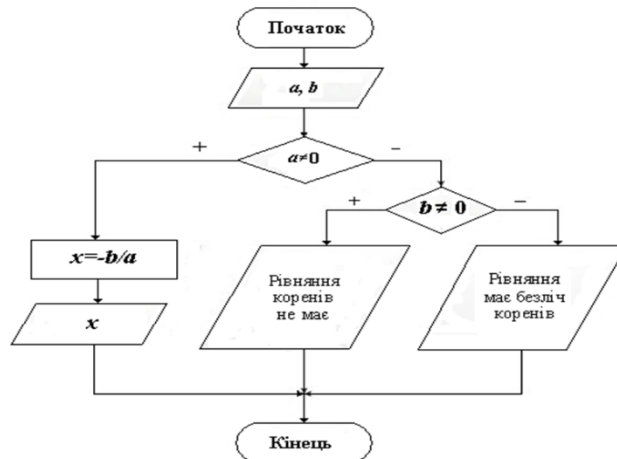


Рис. 1. Блок-схема алгоритму розв’язування лінійного рівняння

На основі блок-схеми пишемо програмний код для даної задачі (Рис. 2).

```
a=int(input("Введіть значення a="))
b=int(input("Введіть значення b="))
if a!=0:
    x=-b/a
    print("x=",x)
else:
    if b!=0:
        print("Рівняння коренів не має")
    else:
        print("Рівняння має безліч коренів")
```

Рис. 2. Програмний код для розв’язування лінійного рівняння

Після написання програмного коду і його відлагодження настає етап тестування. Для цього для різних можливих значень параметрів a,b тестується програмний код. Такими можливими значення параметрів можуть бути наприклад:

1. a=3,b=6;
2. a=0,b=5;
3. a=0,b=0.

Результат тестування програмного коду наведено на Рис. 3.

```
===== RESTART: C:/Python34/linear_eq.py =====
Введіть значення a=3
Введіть значення b=6
x= -2.0
>>>
===== RESTART: C:/Python34/linear_eq.py =====
Введіть значення a=0
Введіть значення b=5
Рівняння коренів не має
>>>
===== RESTART: C:/Python34/linear_eq.py =====
Введіть значення a=0
Введіть значення b=0
Рівняння має безліч коренів
```

Рис. 3. Тестування програмного коду для розв’язування лінійного рівняння

Розглянемо розв’язування квадратного рівняння

Приклад 2. Скласти програму, за якою отримуються розв’язки квадратного рівняння $ax^2+bx+c=0$ з дійсними коефіцієнтами a ($a \neq 0$), b, c.

Опишемо алгоритм словесно-аналітично.

1. Ввести значення a,b,c.
2. Обчислити дискримінант $D=b^2-4ac$.
3. Якщо $D \geq 0$, то виконати п.4, інакше – п.7.
4. Обчислити корені $x_1=(-b-\sqrt{D})/(2a)$, $x_2=(-b+\sqrt{D})/(2a)$.
5. Вивести значення x_1, x_2 .
6. Перейти до пункту 8.
7. Вивести повідомлення «Рівняння не має дійсних коренів».
8. Кінець.

На основі словесно-аналітичного опису алгоритму складемо блок-схему (Рис. 4):

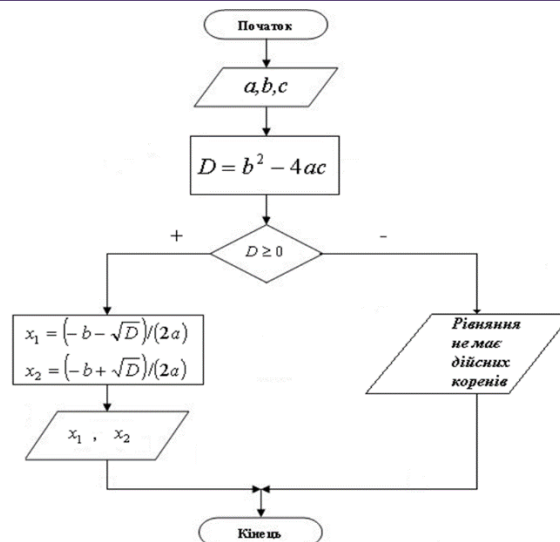


Рис. 4. Блок-схема алгоритму розв’язування квадратного рівняння

На основі блок-схеми пишемо програмний код для даної задачі (Рис. 5). Слід зауважити, що перед написанням програмного коду учням необхідно нагадати, що використання функції для обчислення квадратного кореня (math.sqrt), треба під’єднати математичну бібліотеку math (import math).

```
import math
a=int(input("Введіть значення a="))
b=int(input("Введіть значення b="))
c=int(input("Введіть значення c="))
D=b**2-4*a*c
if D>=0:
    x1=(-b-math.sqrt(D))/(2*a)
    x2=(-b+math.sqrt(D))/(2*a)
    print("x1=",x1, " x2=",x2)
else:
    print("Рівняння не має дійсних коренів")
```

Рис. 5. Програмний код для розв’язування квадратного рівняння

Програмний код тестуємо для можливих значень коефіцієнтів a (a≠0),b,c. Такими значеннями можуть бути наприклад:

1. a=1,b=-5,c=6 – рівняння має два дійсних різних корені;
2. a=1,b=2,c=1 – рівняння має два дійсних рівних корені;
3. a=1,b=1,c=1 – рівняння не має дійсних коренів.

Результат тестування програмного коду наведено на Рис. 6.

```
===== RESTART: C:/Python34/quadratic_eq.py =====
Введіть значення a=1
Введіть значення b=-5
Введіть значення c=6
x1= 2.0 x2= 3.0
>>>
===== RESTART: C:/Python34/quadratic_eq.py =====
Введіть значення a=1
Введіть значення b=2
Введіть значення c=1
x1= -1.0 x2= -1.0
>>>
===== RESTART: C:/Python34/quadratic_eq.py =====
Введіть значення a=1
Введіть значення b=1
Введіть значення c=1
Рівняння не має дійсних коренів
```

Рис. 6. Тестування програмного коду для розв’язування квадратного рівняння

Після тестування програмного коду (тобто перевірки правильності роботи) учням ставиться питання: «Як буде працювати програмний код за умови a=0?». Далі перевіряємо роботу програмного коду за умови a=0 (Рис. 7).

```
===== RESTART: C:/Python34/quadratic_eq.py =====
Введіть значення a=0
Введіть значення b=2
Введіть значення c=3
Traceback (most recent call last):
  File "C:/Python34/quadratic_eq.py", line 7, in <module>
    x1=(-b-math.sqrt(D))/(2*a)
ZeroDivisionError: float division by zero
```

Рис. 7. Результат програмного коду за умови a=0

Як видно з результату, отримується повідомлення про помилку у виразі $x_1 = \frac{-b - \sqrt{D}}{2a}$ – «ділення на нуль» (ZeroDivisionError: float division by zero).

Однією з властивостей алгоритмів є масовість, тобто алгоритм повинен розроблятися для цілого класу задач, що відрізняються тільки вхідних даних. Тому учням самостійно пропонується скласти алгоритм та програмний код для розв'язування рівняння $ax^2 + bx + c = 0$ в загальному випадку, тобто враховуючи і значення $a=0$. При цьому їм пояснюється, що для розв'язування такої задачі необхідно об'єднати в єдине ціле дві вище наведені задачі, тобто розв'язування лінійного та квадратного рівнянь, що задані своїми коефіцієнтами. Очікуваний результат – словесно-аналітичний опис алгоритму, блок-схема та програмний код з його тестування при різних можливих наборах значень коефіцієнтів a, b, c .

Зауважимо, що задачі з параметрами розвивають в учнів, логічне мислення, вміння аналізувати, формуються навички дослідницької діяльності.

Крім того, з даної теми учням можна пропонувати задачі такого типу:

- 1) Визначити, чи заданий рік є високосним. Попередньо учням пропонується знайти, яким умовам повинен відповідати високосний рік.
- 2) Визначити, чи однакового кольору дві клітинки на шаховій дошці. Клітинка однозначно визначається номером (від 1 до 8) рядка і стовпця.

У мові Python є можливість багатоальтернативного вибору. Це можна продемонструвати на такому прикладі.

Приклад 3. У п'ятницю у десятому класі є шість уроків: 1 – алгебра, 2 – українська мова, 3 – інформатика, 4 – географія, 5 – біологія, 6 – фізкультура. Необхідно скласти програму визначення назви уроку за його номером.

Очевидно, що таке завдання зручніше виконувати, використовуючи багатоальтернативний вибір. Програмний код наведено на Рис. 8.

```

номер=int(input("Введіть номер уроку "))
if номер==1:
    print("Урок алгебра")
elif номер==2:
    print("Урок українська мова")
elif номер==3:
    print("Урок інформатика")
elif номер==4:
    print("Урок географія")
elif номер==5:
    print("Урок біологія")
elif номер==6:
    print("Урок фізкультура")
else:
    print("Такого номеру уроку нема")
    
```

Рис. 8. Реалізація багатоальтернативного вибору

Тестування програмного коду наведено на Рис. 9.

```

===== RESTART: C:/Python34/uroky.py =====
Введіть номер уроку 5
Урок біологія
>>>
===== RESTART: C:/Python34/uroky.py =====
Введіть номер уроку 2
Урок українська мова
>>>
===== RESTART: C:/Python34/uroky.py =====
Введіть номер уроку 7
Такого номеру уроку нема
    
```

Рис. 9. Тестування програмного коду на багатоальтернативний вибір

Вивчення циклічних структур ми рекомендуємо проводити на побудові алгоритмів ігрових задач та їх програмної реалізації. Наприклад, гра «Вгадай число» чи «Двадцять одне». Як показує досвід, навчання на таких задач підвищує рівень мотивації в учнів.

Зокрема, роботу циклічної структури з післяумовою можна продемонструвати на прикладі гри «Вгадай число» (Ковалюк, 2005), зміст якої є таким.

Приклад 6. Програма генерує випадкове ціле число з деякого діапазону (наприклад, від 1 до 1000). Користувач намагається його відгадати, вводячи його з клавіатури. Якщо число вгадане, то програма виводить повідомлення про це і завершує роботу, інакше – спроби відгадати число повторюються. На початку виконання програми встановлюється певний «призовий фонд» (наприклад, 100), що зменшується з кожної невдалою спробою (наприклад, на 10). Гра закінчується у двох випадках: гравець відгадав число або «призовий фонд» дорівнює нулю (Ковалюк, 2005).

Розв'язування. Очевидно, що процес відгадування числа є циклічним. Тут доцільно використовувати цикл з післяумовою, оскільки принаймні повинна відбутися одна спроба вгадування. Зауважимо, що у мові Python відсутній такий оператор циклу. Проте його можна реалізувати, використовуючи цикл з передумовою так:

```

while True:
    тіло циклу
    if <умова> break:
    
```

Змінні, які будемо використовувати у програмному коді: prize – призовий фонд, y – випадкове ціле число, згенероване комп'ютером, penalty – штраф, x – число, яке вводиться гравцем. Зауважимо, що для генерації цілого випадкового числа від 1 до 1000 використовується функція randint(1,1000) з бібліотеки random (import random).

Програмний код гри «Вгадай число» наведено на рис.10.

```
import random
penalty=10
y=random.randint(1,1000)# генерація випадкового цілого числа від 1 до 1000
prize=100
while True:
    x=int(input("Введіть число від 1 до 1000: "))
    if x>y:
        print("Неправильно, введіть менше число")
        prize=prize-penalty
    else:
        if x<y:
            print("Неправильно, введіть більше число")
            prize=prize-penalty
        else:
            print("Вітаємо! Ви виграли! Ваш виграш становить ",prize)
            break
    if prize==0:
        print("На жаль, Ви програли. Спробуйте ще")
        print("Загадане число: ",y)
        break
input()
```

Рис. 10. Програмний код гри «Вгадай число»

Результати тестування програмного коду гри «Вгадай число» наведено на рис 11. На рис 11,а наведено випадок виграшу гравцем, а на рис 11,б – програшу.

```
===== RESTART: C:/Python34/ugade
Введіть число від 1 до 1000: 500
Неправильно, введіть менше число
Введіть число від 1 до 1000: 250
Неправильно, введіть менше число
Введіть число від 1 до 1000: 125
Неправильно, введіть менше число
Введіть число від 1 до 1000: 65
Неправильно, введіть менше число
Введіть число від 1 до 1000: 35
Неправильно, введіть менше число
Введіть число від 1 до 1000: 18
Неправильно, введіть менше число
Введіть число від 1 до 1000: 9
Вітаємо! Ви виграли! Ваш виграш становить 40
|
```

а)

```
===== RESTART: C:/Py
Введіть число від 1 до 1000: 576
Неправильно, введіть менше число
Введіть число від 1 до 1000: 423
Неправильно, введіть менше число
Введіть число від 1 до 1000: 345
Неправильно, введіть менше число
Введіть число від 1 до 1000: 234
Неправильно, введіть менше число
Введіть число від 1 до 1000: 123
Неправильно, введіть більше число
Введіть число від 1 до 1000: 145
Неправильно, введіть більше число
Введіть число від 1 до 1000: 157
Неправильно, введіть більше число
Введіть число від 1 до 1000: 167
Неправильно, введіть більше число
Введіть число від 1 до 1000: 178
Неправильно, введіть більше число
Введіть число від 1 до 1000: 188
Неправильно, введіть більше число
На жаль, Ви програли. Спробуйте ще
Загадане число: 232
```

б)

Рис. 11. Тестування програмного коду гри «Вгадай число»

ВИСНОВКИ ТА ПЕРСПЕКТИВИ ПОДАЛЬШОГО ДОСЛІДЖЕННЯ

Отже, узагальнивши відомості, зваживши всі переваги і недоліки, ми прийшли до висновку, що Python наразі є оптимальною мовою програмування для вивчення основ алгоритмізації у шкільному курсі інформатики, тому що:

- має простий синтаксис, програмний код легко читається;
- це об'єктно-орієнтована мова програмування високого рівня, призначена для розв'язування найрізноманітніших завдань;
- є кросплатформенною мовою, що дозволяє створювати програми, які будуть працювати у всіх операційних системах.
- є бібліотеки готових процедур для використання у своїх програмах. Це дозволяє створювати складні програми швидко.
- підтримує різні парадигми програмування.

Мова Python має потужну стандартну бібліотеку, яку користувач може розширювати власними бібліотеками й бібліотеками інших користувачів.

У статті ми навели деякі методичні підходи для вивчення алгоритмічної структури розгалуження та повторення, зокрема реалізацію циклу з післяумовою. На конкретних прикладах продемонстрували реалізацію окремих алгоритмів мовою Python. Звертаємо увагу, що з використанням мови Python такі класичні алгоритми як пошук мінімального (чи

максимального) елемента, обміну значеннями між змінними втрачають своє значення. Відповідно зміст навчання треба наповнювати іншими завданнями. На нашу думку, важливим проблемою вивчення основ алгоритмізації і програмування з використанням мови Python є підбір таких завдань, щоб в учнів якомога рідше виникло питання «навіщо нам вивчати алгоритм, якщо це можна зробити, використовуючи одну-дві функції мови Python?».

Подальші дослідження будуть спрямовані на методиці навчання опрацювання складних структур даних, зокрема списків, кортежів, словників, рядків, засобами мови Python.

Список використаних джерел

1. Базурін В. М. Порівняльний аналіз середовищ програмування мовою Python. Новітні комп'ютерні технології. Кривий Ріг : Видавничий центр ДВНЗ «Криворізький національний університет», 2018. Том XVI. С. 281-292.
2. Інформатика для 10-11 класів (профільне навчання). URL: <https://mon.gov.ua/storage/app/media/zagalna%20serednya/programy-10-11-klas/2018-2019/01/10-11-profilniy-riven.docx> (дата звернення 12.04.2020)
3. Інформатика. Навчальна програма вибірково-обов'язкового предмету для учнів 10-11 класів загальноосвітніх навчальних закладів (рівень стандарту). URL: <https://mon.gov.ua/storage/app/media/zagalna%20serednya/programy-10-11-klas/2018-2019/informatika-standart-10-11.docx> (дата звернення 12.04.2020)
4. Морзе Н. В., Барна О. В., Вембер В. П. Інформатика: підр. для 8 кл. загальноосвіт. навч. Закладів.К. : УОВЦ «Оріон», 2016. 240 с.
5. Морзе Н. В., Барна О. В., Вембер В. П. Інформатика: підр. для 9 кл. загальноосвіт. навч. Закладів.К. : УОВЦ «Оріон», 2016. 208 с.
6. Морзе Н.В., Барна О.В. Інформатика. Підручник для 7 кл. закладів загальної середньої освіти. Київ: УОВЦ «Оріон», 2020. 176 с.
7. Руденко В.Д., Речич Н.В., Потієнко В. О. Інформатика (профільний рівень) : підруч. для 10 кл. закл. загал, серед, освіти. Харків : Вид-во «Ранок», 2019. 256 с.
8. Руденко В.Д., Речич Н.В., Потієнко В. О. Інформатика (профільний рівень) : підруч. для 11 кл. закл. загал, серед, освіти. Харків : Вид-во «Ранок», 2019. 256 с.
9. Юрченко А.О., Семеніхіна О.В., Хворостіна Ю.В., Удовиченко О.М., Петренко С.І. Навчання програмувати в старшій школі крізь призму чинних навчальних програм. Фізико-математична освіта. 2019. Випуск 2(20). Ч. 2. С. 48-55.
10. Яку мову програмування вивчати у школі. Комп'ютер у школі та сім'ї №8, 2013. С. 9-18.
11. Ривкінд Й.Я., Лисенко Т.І., Чернікова Л.А., Шакотько В.В. Інформатика : підруч. для 8-го кл. закл. заг. серед. освіти. К.: Генеза, 2021. 256 с.
12. TIOBE Index. URL: <https://www.tiobe.com/tiobe-index/> (дата звернення: 26.09.2021).
13. Ковалюк Т.В. Основи програмування. К.:Видавнича група BHV, 2005. 384 с.

References

1. Bazurin, V.M. (2018) Comparative analysis of Python programming environments. The latest computer technology. Kryvyi Rih: Kryvyi Rih National University Publishing Center. Volume XVI. Pp. 281-292. [in Ukrainian].
2. Informatics for 10-11 grades (profile training). (2020) URL: <https://mon.gov.ua/storage/app/media/zagalna%20serednya/programy-10-11-klas/2018-2019/01/10-11-profilniy-riven.docx> (access date 12.04. 2020). [in Ukrainian].
3. Informatics. Curriculum of elective-compulsory subject for students of 10-11 grades of secondary schools (standard level). (2020) URL: <https://mon.gov.ua/storage/app/media/zagalna%20serednya/programy-10-11-klas/2018-2019/informatika-standart-10-11.docx> (accessed 12.04.2020). [in Ukrainian].
4. Morse, N.V., Barna, O.V., Wember, V.P. (2016). Informatics: ed. for 8 classes. general education. textbook Institutions.K. : УОВЦ «Оріон». 240 с. [in Ukrainian].
5. Morse, N.V., Barna, O.V., Wember, V.P. (2016). Informatics: ed. for 9 classes. general education. textbook Institutions.K. : УОВЦ «Оріон». 208 с. [in Ukrainian].
6. Morse, N.V., Barna, O.V. (2020). Computer Science. Textbook for 7 classes. general secondary education institutions. Kyiv: UORC Orion. 176 p. [in Ukrainian].
7. Rudenko, V.D., Rechich, N.V., Potienko, V.O. (2019). Informatics (profile level): textbook. for 10 classes. lock general, among, education. Kharkiv: Ranok Publishing House. 256 p. [in Ukrainian].
8. Rudenko, V.D., Rechich, N.V., Potienko, V.O. (2019). Informatics (profile level): textbook. for 11 classes. lock general, among, education. Kharkiv: Ranok Publishing House. 256 p. [in Ukrainian].
9. Yurchenko, A.O., Semenikhina, O.V., Khvorostina, Y.V., Udovychenko, O.M., Petrenko, S.I. (2019). Learning to program in high school through the prism of existing curricula. Physical and mathematical education. Issue 2 (20). Part 2. pp. 48-55. [in Ukrainian].
10. What programming language to study at school. (2013). Computer in school and family №8. P.9-18. [in Ukrainian].
11. Rivkind, Y.Y., Lysenko, T.I., Chernikova, L.A., Shakotko, V.V. (2021). Informatics: textbook. for the 8th grade. lock head among, education. K. : Genesis. 256 p. [in Ukrainian].
12. TIOBE Index. URL: <https://www.tiobe.com/tiobe-index/> (date of application: 26.09.2021). [in English].
13. Kovalyuk, T.V. (2005). Basics of programming. K.: BHV Publishing Group. 384 p. [in Ukrainian].

METHODICAL ASPECTS OF STUDYING THE FUNDAMENTALS OF ALGORITHMIZATION AND PROGRAMMING LANGUAGE PYTHON SCHOOL COURSE IN INFORMATICS IN HIGH SCHOOL**Taras Kobylnyk¹, Uliana Kohut², Volodymyr Zhydyk²**¹Lviv Polytechnic National University, Ukraine²Drohobych Ivan Franko State Pedagogical University, Ukraine

Abstract. *The article investigates the problems of studying the elements of programming in the school course of computer science.*

Formulation of the problem. *An important pedagogical problem is the formation and development of the algorithmic skills of students, the improvement of their professional training. The problems of improving the methodological system of teaching computer science in general secondary school require further research in order to find more effective forms, methods and means of teaching. It is necessary to ensure the formation in students of the competencies necessary for a conscious choice of the future profile of education. Students should understand the relationship between theoretical and applied informatics, know how to use various types of information technology in professional activities. And also the advantages of the Python language in teaching informatics and the features of their pedagogical use in teaching the basics of algorithms are determined.*

Materials and methods. *To achieve this goal, a set of methods was used: a systematic analysis of scientific works covering the problem of research, studying the domestic experience of studying programming elements in a school computer science course; generalization and substantiation of the theoretical foundations of using the PYTHON programming language in the school computer science course.*

Results. *The main characteristics of the Python programming language are considered and the possibilities of using this language in the process of teaching the basics of algorithmization in the school computer science course are presented. The study of the basics of algorithms and programming is one of the means of forming the operational style of thinking. Knowledge of this section presupposes the ability to formalize the task, determine the connections between the parts in it, choose the most effective algorithm for solving the problem, and the ability to correctly interpret and analyze the results obtained.*

Conclusions. *The Python language is one of the most popular programming languages. It finds its application in various fields. According to the labor market, there is a need for specialists who could create software products in the Python language. The institutions of general secondary education did not remain aloof from this process. School textbook authors are starting to incorporate Python into their content. Thus, there is a need for the development of teaching materials for the study of the elements of algorithmization and programming in Python. The methodological significance of this section of the school computer science course lies in the disclosure of the importance of algorithms. Fundamentals of Algorithmization and Programming is one of the main content lines of the computer science course in basic school.*

Key words: *institution of general secondary education, school course in computer science, the basics of algorithms, programming, programming language Python.*

